
Map2Tsym: A Map File to TextSym File Conversion Utility Version 1.02.00

An Application Note

www.arium.com

American Arium's Map2Tsym utility (v1.02.00) is a Win32 console application that uses map files to generate TextSym files containing address and symbol information. This allows WinDb to support symbolic debugging and accept symbol names when specifying memory locations to examine code, view memory, or set breakpoints. As a Win32 console application, it executes under Microsoft Windows XP/2000/NT/ME/98/95. It will not operate under Win 3.x or stand-alone MS-DOS.

Map2Tsym accepts two map file formats: the MAPCONV-002 file generated by the Microsoft link program, and the MAPCONV file generated by the Phoenix MAPCONV program.

Map2Tsym is invoked from the command line, batch file, or make utility as follows:

`Map2Tsym source [destination]`

Where: *source* is the pathname of the input map file

destination is the pathname of the output TextSym file

(NOTE: If *destination* is not specified, the *source* file name is used with the '.sym' extension);

If long file or path names are used, they must be contained within double quotes as shown in the following example that uses a long destination name:

Map2Tsym bios.map "c:\Program Files\American Arium\WinDb\symbols.sym"

This utility may be used as part of the tool chain that produces the executable file. To facilitate such use, Map2Tsym produces the following exit codes that can be tested by batch commands to verify its success.

Exit Code	Condition
0	Successful
1	Could not open source file
2	Could not open destination file
3	Could not read from source file
4	Could not write to destination file
5	No symbols found or any other problem

Once the TextSym file has been generated by Map2Tsym, it can be loaded into WinDb using the file menu on the menu bar (File | Program | Load Program). After loading the TextSym file, symbol names can be entered in lieu of literal address values to specify locations.

Symbolic Text File Format (TextSym)

TextSym is a simple text file format used to convey symbolic debug information. The output file from the Map2Tsym utility conforms to this specification.

Field Separator

Each field is separated by the vertical bar ('|') character. White space around the bar is optional. All leading and trailing white spaces between fields are ignored.

Signature

The first line of this text file contains a signature and version information. The "TEXTSYM format" string and version number must be as shown. White space will be ignored. If a valid signature is not found, the load will abort.

```
TEXTSYM format | V1.1 <eol>
```

Debug Information

Debug information for each symbol is specified on a separate line as specified below:

GLOBAL/LOCAL	Offset Value	CODE/DATA	Symbol Name	Object Size
--------------	--------------	-----------	-------------	-------------

Where:

GLOBAL/LOCAL	Usage is tool dependent. If symbol is specified as GLOBAL, then it must be unique within this module - no duplication is allowed. Some tools may ignore symbols marked as LOCAL.			
Offset Value	64-bit hex value treated as an unsigned number. The offset value is added to the address where the symbol file is loaded.			
CODE/DATA	A required keyword. When the debug tool forms an IA-64 symbolic address, this field is used to determine whether the resulting symbol has a data address or an execution address. This field is ignored when reading IA-32 symbols.			
Symbol Name	A contiguous ASCII string of characters that are legal to identify a variable/function name in C/C++. Symbol names are case sensitive. Length is not restricted but limited by the debug tool that consumes it.			
Object Size	The size of a data object in bytes. This field is optional. It is in bytes for code and data symbols. For IA-64 code, it will be interpreted to point to whole bundles. This field is not allowed in version 1.0 and is optional in version 1.1. Both versions 1.0 and 1.1 are currently supported in WinDb.			

Example:

```
TEXTSYM format | V1.0 <eol>
GLOBAL | 00000000c0000000 | CODE | ENTER_RESET <eol>
GLOBAL | 0000000000000430 | DATA | OStypeFound <eol>
LOCAL | 0000000000001234 | CODE | BAR <eol>
GLOBAL | 0000000000001238 | DATA | FOO | 4 <eol>
```

